OpenMP

Category: Porting & Developing Applications

DRAFT

This article is being reviewed for completeness and technical accuracy.

OpenMP is a portable, scalable model that gives shared-memory parallel programmers a simple and flexible interface for developing parallel applications for various platforms.

Intel version 11.x compilers support OpenMP spec-3.0 while 10.x compilers support spec-2.5.

Building OpenMP Applications

The following Intel compiler options can be used for building or analyzing OpenMP applications:

- *-openmp*

  Enables the parallelizer to generate multithreaded code based on OpenMP directives. The code can be executed in parallel on both uniprocessor and multiprocessor systems. The *-openmp* option works with both *-O0* (no optimization) and any optimization level of *-O*. Specifying *-O0* with *-openmp* helps to debug OpenMP applications.

  Note that setting *-openmp* also sets *-automatic*, which causes all local, non-SAVEd variables to be allocated to the run-time stack, which may provide a performance gain for your applications. However, if your program depends on variables having the same value as the last time the routine was invoked, your program may not function properly. If you want to cause variables to be placed in static memory, specify option *-save*. If you want only scalar variables of certain intrinsic types (integer, real, complex, logical) to be placed on the run-time stack, specify option *-auto-scalar*.
- *-assume cc_omp* or *-assume nocc_omp*

  -assume cc_omp enables conditional compilation as defined by the OpenMP Fortran API. That is, when "!$space" appears in free-form source or "c$spaces" appears in column 1 of fixed-form source, the rest of the line is accepted as a Fortran line.

  -assume nocc_omp tells the compiler that conditional compilation as defined by the OpenMP Fortran API is disabled unless option -openmp (Linux) or /Qopenmp

(Windows) is specified.

- *-openmp-lib legacy* or *-openmp-lib compat*

  Choosing *-openmp-lib legacy* tells the compiler to use the legacy OpenMP run-time library (*libguide*). This setting does not provide compatibility with object files created using other compilers. This is the default for Intel version 10.x compilers.

  Choosing *-openmp-lib compat* tells the compiler to use the compatibility OpenMP run-time library (*libiomp*). This is the default for Intel version 11.x compilers.

  On Linux systems, the compatibility Intel OpenMP run-time library lets you combine OpenMP object files compiled with the GNUgcc or gfortran compilers with similar OpenMP object files compiled with the Intel C/C++ or Fortran compilers. The linking phase results in a single, coherent copy of the run-time library.

  You cannot link object files generated by the Intel® Fortran compiler to object files compiled by the GNU Fortran compiler, regardless of the presence or absence of the *-openmp* (Linux) or /Qopenmp (Windows) compiler option. This is because the Fortran run-time libraries are incompatible.

  NOTE: The compatibility OpenMP run-time library is not compatible with object files created using versions of the Intel compiler earlier than 10.0.

- *-openmp-link dynamic* or *-openmp-link static*

  Choosing *-openmp-link dynamic* tells the compiler to link to dynamic OpenMP run-time libraries. This is the default for Intel version 11.x compilers.

  Choosing *-openmp-link static* tells the compiler to link to static OpenMP run-time libraries.

  Note that the compiler options *-static-intel* and *-shared-intel* have no effect on which OpenMP run-time library is linked.

  Note that this option is only available for newer Intel compilers (version 11.x).
- *-openmp-profile*

  Enables analysis of OpenMP applications. To use this option, you must have Intel(R) Thread Profiler installed, which is one of the Intel(R) Threading Tools. If this threading tool is not installed, this option has no effect.

  Note that Intel Thread Profiler is not installed on Pleiades.

- *-openmp-report[n]*

Controls the level of diagnostic messages of the OpenMP parallelizer. n=0,1,or 2.

- *-openmp-stub*

  Enables compilation of OpenMP programs in sequential mode. The OpenMP directives are ignored and a stub OpenMP library is linked.

OpenMP Environment Variables

There are a few OpenMP environment variables one can set. The most commonly used are:

- *OMP_NUM_THREADS num*

  Sets number of threads for parallel regions. Default is 1 on Pleiades. Note that you can use *ompthreads* in the PBS resource request to set values for OMP_NUM_THREADS. For example:

```
%qsub -I -lselect=1:ncpus=4:ompthreads=4
Job 991014.pbspl1.nas.nasa.gov started on Sun Sep 12 11:33:06 PDT 2010
...
PBS r3i2n9> echo $OMP_NUM_THREADS
4
PBS r3i2n9>
```

- *OMP_SCHEDULE type[,chunk]*

  Sets the run-time schedule type and chunk size. Valid OpenMP schedule types are *static, dynamic, guided,* or *auto*. Chunk is a positive integer.

- *OMP_DYNAMIC true* or *OMP_DYNAMIC false*

  Enables or disables dynamic adjustment of threads to use for parallel regions.

- *OMP_STACKSIZE size*

  Specifies size of stack for threads created by the OpenMP implementation. Valid values for size (a positive integer) are *size, size*B*, size*K*, size*M*, size*G. If units B, K, M or G are not specified, size is measured in kilobytes (K).

  Note that this feature is included in OpenMP spec-3.0, but not in spec-2.5.

Note that Intel also provides a few additional environment variables. The most commonly used are:

- *KMP_AFFINITY type*

Binds OpenMP threads to physical processors. Avaiable *type*: *compact, disabled, explicit, none, scatter*. For more information on the various types, see this Intel web page.

```
There is a conflict between KMP_AFFINITY in Intel 11.x runtime
library and dplace, causing all threads to be placed on a
single CPU when both are used. It is recommended that
KMP_AFFINITY be set to disabled when using dplace.
```

- *KMP_MONITOR_STACKSIZE*

  Sets stacksize in bytes for monitor thread.

- *KMP_STACKSIZE*

  Sets stacksize in bytes for each thread.

For more information, please see the official OpenMP web site.

Article ID: 209
Last updated: 21 Jun, 2011
Computing at NAS -> Porting & Developing Applications -> OpenMP
http://www.nas.nasa.gov/hecc/support/kb/entry/209/?ajax=1